

## Матрица грейдов python разработчика

	Junior	Middle	Senior	Architect
<b>Теория</b>				
Опыт работы с СУБД, хранилищами данных	Умеет работать с РСУБД на базовом уровне (базовые DDL, DML операции).	Понимает ACID, транзакции. Умеет оценить сложность запросов на базе плана выполнения, построить необходимые индексы. Может спроектировать файловое хранилище для приложения.	Может спроектировать хорошие нормализованные схемы БД, с учетом запросов, которые будут выполняться. Умеет применять представления, хранимые процедуры, триггеры и собственные типы данных. Понимает разницу между кластеризованными и не-кластеризованными индексами. Имеет опыт работы с нереляционными СУБД.	Имеет опыт и навыки проектирования БД и файловых хранилищ для высоконагруженных систем. Опыт работы с кластерами, администрирования и оптимизации производительности СУБД.
Структуры данных и алгоритмы	Может объяснить и использовать на практике массивы, связанные списки, словари и т.д. Может использовать базовые алгоритмы работы с этими структурами (поиск, обход, сортировка)	Понимает и умеет выбирать структуры данных и алгоритм под задачу (временная и пространственная сложность алгоритма). Знает особенности и умеет применять графы( деревья), хеш-таблицы.	Знание сложных структур данных (В-дерево, Биномиальная куча и Фибоначчиевская куча, АВЛ-дерево, Красно-чёрное дерево, Косое дерево, Список с пропусками, TRIE-структуры и т.д.).	Может распознать и написать решение динамическим программированием, хорошо знает алгоритмы на графах, хорошо знает численные методы, может идентифицировать проблемы класса NP.
Навыки	Может создать программу/систему по	Отлично владеет ООП, может свободно	Владеет навыками проектирования информационных систем, выходя за	Имеет опыт проектирования

проектирования информационных систем	предлагаемым шаблонам/гайдам. Знает про ООП, DRY на уровне статей в Википедии.	рассуждать о плюсах и минусах. Владеет основными шаблонами проектирования, может их распознать. Может спроектировать несложную ненагруженную систему, способную стабильно работать и развиваться.	пределы шаблонов. Понимает применимость шаблонов, может обосновать их использование/неиспользование. Знание UML на уровне, позволяющем создавать модели прикладных систем.	высоконагруженных программных комплексов, состоящих из нескольких систем, возможно, с едиными системами управления доступом, шинами событий, данных.
Веб-протоколы	HTTP	SOAP/WSDL, REST, Websocket	-	-
Криптография	-	-	Открытый/закрытый ключ/Цифровая подпись/Сертификат	-

## Практика

Знание технологий и платформ	Знает на начальном уровне требуемый стек технологий и фреймворки. Знаком (возможно без личного опыта) с наиболее популярными компонентами/фреймворками в своем стеке технологий.	Знает альтернативные технологии, сильные/слабые стороны. Уверенно владеет основными используемыми фреймворками/компонентами, пробовал другие, знает их особенности. Имеет опыт работы с нагруженными системами. Понимает как происходит сборка приложения. Понимает как происходит исполнение приложения, управление памятью, обращение к ресурсам среды исполнения.	Понимает как работает весь "программный стек": железо/среда исполнения, компиляция(jit)/линковка/интерпретация, управление памятью(сборка мусора, куча, стек, адресация памяти). Понимает работу с многопоточностью, синхронизацией потоков. Писал компоненты/фреймворки используемые в компании/сообществе.	-
------------------------------	--	--	--	---

Python	"Практическое введение в Python для Django" (книга "Django. Разработка веб-приложений на Python", Джефф Форсье, Пол Биссекс, Уэсли Чан)	глава "I. Язык программирования Python" от и до, разделы 12-15 из II главы (книга "Python. Подробный справочник" Автор: David Beazley)	Потоки и многозадачность, Работа с сетью и сокеты. разделы 16-25 из II главы (книга "Python. Подробный справочник" Автор: David Beazley)	-
Django	"Первые шаги" (документация Django)	Официальная документация к Django "от и до"	-	-
SQL (в postgres)	SQL Syntax , Data Definition, Data Manipulation	Queries, Data Types, Functions and Operators, Type Conversion, Full Text Search, Indexes, Performance Tips	Теоретические знания о Triggers, The Rule System, Procedural Languages - что это и в каких случаях может быть применено. Может при необходимости быстро изучить и применить на практике.	-
Deploy (gunicorn/uwsgi/nginx)	-	Setuptools, pyup, wheels, sdist. Типовая инструкция по установке приложения на linux и gunicorn.	Конфигурация и особенности gunicorn, rabbitMq, redis, memcache. Методы и инструменты мониторинга и анализа работы веб-приложения.	-
Deploy (postgres)	-	-	pg_badger, системные представления, инструменты мониторинга	-
Инструменты разработки	Владеет базисом инструмента разработки/автотестирования, может работать с VCS, может запускать сборки в CI.	Полностью владеет инструментами разработки, принятыми в производственном подразделении, в том числе IDE+расширения (в том числе инструменты профилирования, декомпилирования исходных кодов, автотестов), VCS (понимает принципы работы,	Использует набор инструментов, позволяющий максимально эффективно решать задачи (включая нетиповые). Может обосновать выбор этого набора. Может использовать и создавать расширения инструментов.	-

		ветвления, хранения исходников), CI (может настраивать сборку).		
Git	Основы Git	Ветвление в Git, Распределённый Git	Перезапись истории, Git изнутри	-
IDE	-	практические вопросы по знанию IDE, фичи ускоряющие работу разработчика	-	-